

Propriétés non-fonctionnelles : annotation des processus et gouvernance des services

« Chaque individu apporte au monde sa contribution unique. »

Jack Kornfield

III.1. INTRODUCTION	58
III.2. PETALS BPM-NFR : PLATEFORME DE MODELISATION ET D'ANNOTATION NON-FONCTIONNELLE DES PROCESSUS.....	60
III.2.1. Architecture générale de Petals BPM-NFR	60
III.2.2. Modélisation des processus métier	61
III.2.3. Annotation non-fonctionnelle des processus BPMN 2.0	62
III.3. EASIERGOV-NFR : GOUVERNANCE SOA POUR LA GESTION DES SERVICES ET DES PROPRIETES NON-FONCTIONNELLES	65
III.3.1. Découverte SOA.....	66
III.3.2. Gestion des services	67
III.3.3. Gestion des propriétés non-fonctionnelles	68
III.3.3.1. Annotation non-fonctionnelle des services	68
III.3.3.2. Modèle de contrat de services Web (<i>WS-Agreement template</i>).....	70
III.3.3.3. Classification des propriétés non-fonctionnelles	71
III.3.3.4. Gestion des propriétés non-fonctionnelles dans EasierGov-NFR	74
III.4. CONCLUSION	76

III.1. Introduction

Dans la littérature, bien que le terme « propriétés non-fonctionnelles » soit utilisé depuis plus de 25 ans et que la majorité ait souligné l'importance de l'utilisation de ces propriétés, il n'existe toujours pas un consensus pour les définir, les obtenir et les valider [Glinz, 2007]. Ce manque critique pour la réussite des processus et applications de l'entreprise est peut être dû à plusieurs facteurs ([Rosa et al., 2002] et [Khaled et al., 2005]) :

- les propriétés non-fonctionnelles sont généralement abstraites et la plupart du temps elles sont décrites d'une façon informelle ;
- souvent, il n'existe pas une séparation claire entre les aspects fonctionnels et non-fonctionnels d'un service ;
- les propriétés non-fonctionnelles sont souvent en conflit entre elles et en concurrence les unes avec les autres (par exemple la disponibilité et la performance) ;
- la modélisation des propriétés non-fonctionnelles est souvent complexe.

Afin de combler ce manque et d'atteindre les besoins exprimés dans le chapitre I, nous avons présenté dans le chapitre II des solutions existantes pour la description, la modélisation et la gestion des propriétés non-fonctionnelles (pour les processus métier et pour les services). Nous nous sommes également intéressés à étudier les travaux existants autour : (i) de la gouvernance SOA pour une meilleure gestion et rationalisation des services, (ii) de la réconciliation et la sélection de services selon les propriétés non-fonctionnelles. Cependant, ces travaux présentaient certaines limites (vues dans le chapitre II) qui nous sont essentielles. Ces limites concernent les trois thèmes qui constituent notre étude :

- modélisation des propriétés non-fonctionnelles et annotation des processus métier : il est nécessaire, lors de la phase de modélisation de processus, de permettre aux utilisateurs d'annoter facilement les activités par un ensemble de propriétés non-fonctionnelles. Cet ensemble de propriétés doit être assez large pour qu'ils puissent exprimer leurs besoins métier ;
- rationalisation de la gestion des propriétés non-fonctionnelles par la gouvernance SOA : il est indispensable de gérer l'ensemble des propriétés non-fonctionnelles depuis la phase de la publication des descriptions de services. La gouvernance SOA adresse ce besoin en proposant des concepts visant la modélisation des descriptions de services et la prise en considération de leurs propriétés non-fonctionnelles ;

- réconciliation entre les besoins métier et les services techniques : les approches de réconciliation existantes ne sont pas adaptées dans le cas d'une composition de services (1-N : à une activité métier, nous associons une composition de n services).

La Figure 22 permet de résumer tous ces éléments en détaillant nos besoins, en présentant l'existant et en analysant les limites et manques de ce dernier.

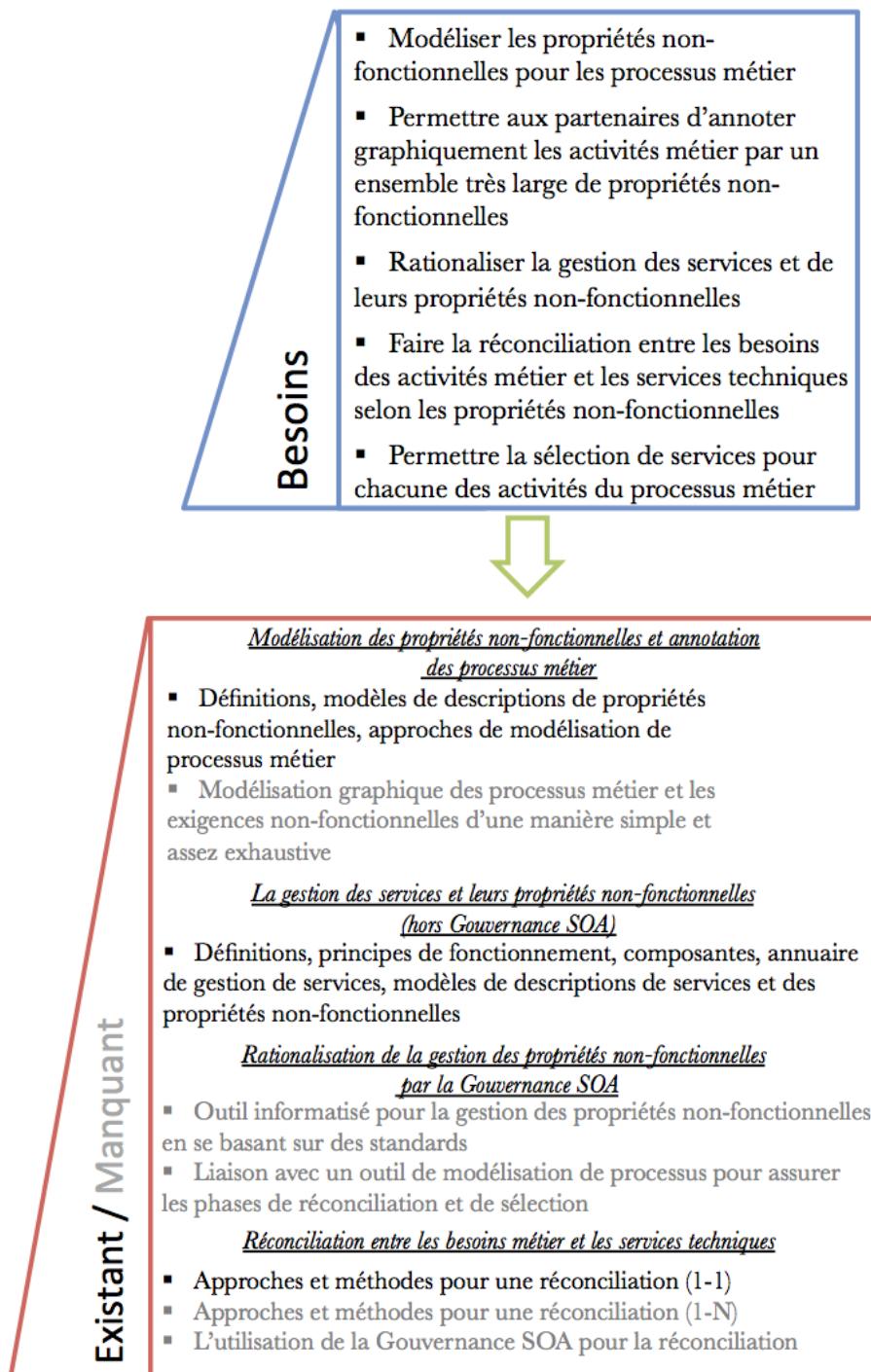


Figure 22 : Analyse croisée : besoins / existant et manquant.

Dans le présent chapitre, comme l'illustre la Figure 23, nous nous intéressons aux deux premiers thèmes de notre problématique de recherche. Nous proposons, dans la Section III.2, un enrichissement de la modélisation graphique des processus métier collaboratifs BPMN 2.0 par l'annotation non-fonctionnelles des activités. Dans la section III.3, nous présentons un cadre permettant de gérer les services et les propriétés non-fonctionnelles au sein du registre de gouvernance SOA. Ce système se base sur des standards afin de garantir principalement l'interopérabilité (aspect essentiel dans un cadre collaboratif).

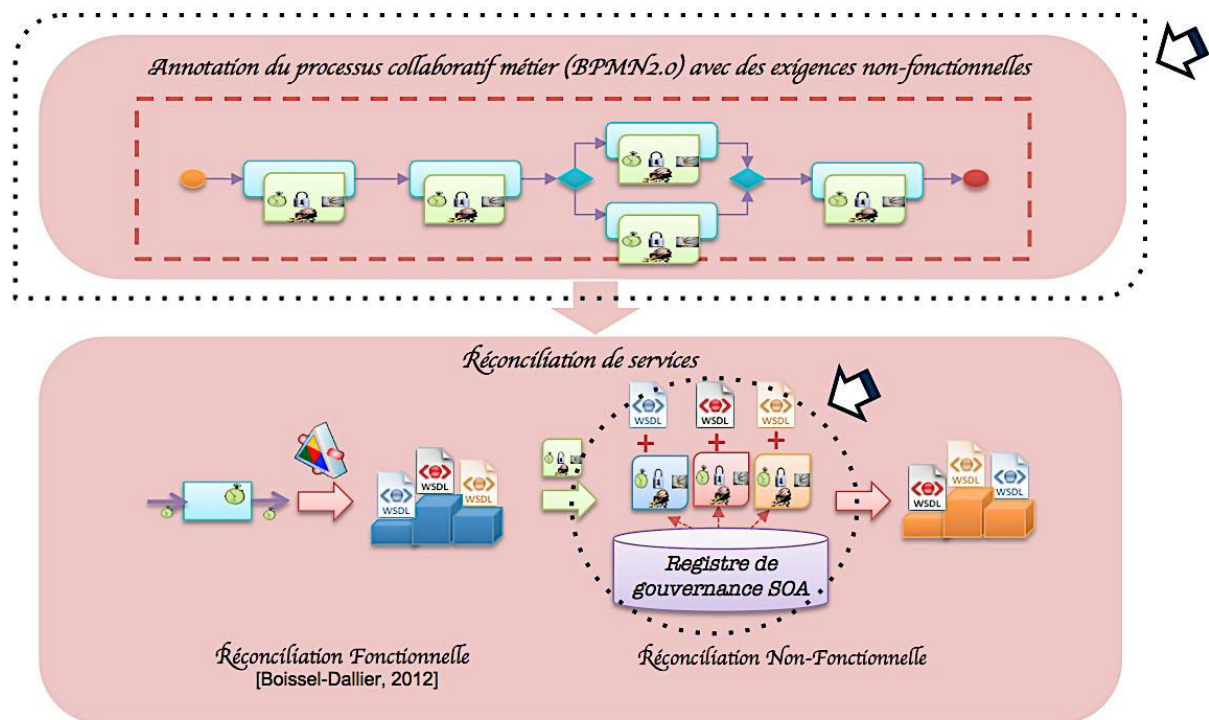


Figure 23 : Modélisation, annotation non-fonctionnelle et gouvernance SOA.

III.2. Petals BPM-NFR : plateforme de modélisation et d'annotation non-fonctionnelle des processus

Dans cette section, nous présentons notre prototype de modélisation et d'annotation graphique des activités métier nommé *Petals BPM-NFR*.

III.2.1. Architecture générale de Petals BPM-NFR

Petals BPM-NFR (Petals Business Process Management - Non-Functional Requirements) est une plateforme qui permet de modéliser des processus métier suivant le standard BPMN 2.0. Cette

plateforme enrichit la modélisation graphique habituelle des processus (établie par Petals BPM) par la possibilité d'annoter les activités métier à l'aide d'exigences non-fonctionnelles. Dans la Figure 24, nous présentons l'architecture générale de la plateforme Petals BPM-NFR.

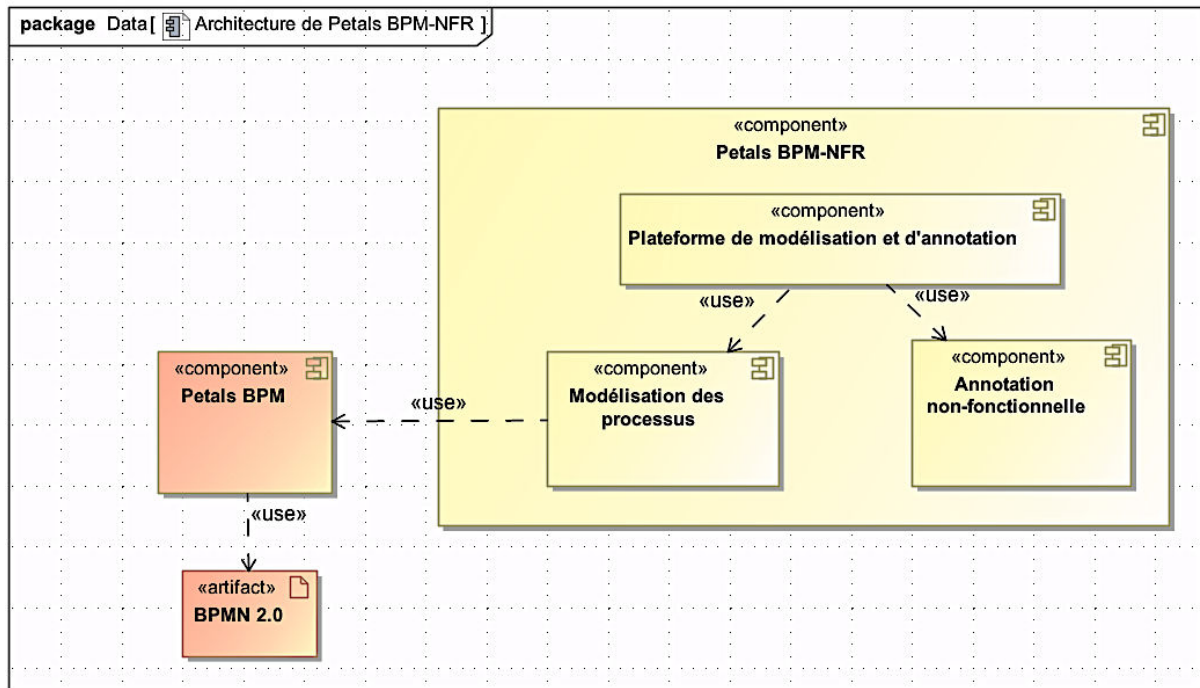


Figure 24 : Architecture générale de Petals BPM-NFR.

Deux composants principaux constituent cette plateforme : (i) «Modélisation des processus » qui permet de modéliser les processus métier collaboratifs et se base sur l'outil de modélisation graphique existant Petals BPM et (ii) « Annotation non-fonctionnelle » qui assure l'annotation non-fonctionnelle des activités métier par des exigences non-fonctionnelles. Dans les sous-sections qui suivent nous détaillons ces deux composants.

III.2.2. Modélisation des processus métier

Nous souhaitons réaliser un éditeur de modélisation de processus métier collaboratifs basé sur Petals BPM et supportant la démarche d'annotation non-fonctionnelle des activités métier.

Petals BPM est un modeleur graphique, open-source, en ligne de processus BPMN 2.0⁴. Dans la Figure 25 nous représentons une capture d'écran de son interface graphique. Il est composé de trois parties : (i) la palette BPMN 2.0 (à gauche) contenant l'ensemble des éléments permettant aux utilisateurs de créer des processus BPMN 2.0, (ii) la zone de modélisation (au centre) qui permet aux utilisateurs de dessiner leurs processus grâce à la palette d'outils, et enfin (iii) la définition des

⁴ Petals BPM : <http://bpmneditor.petalslink.com/>

propriétés (en bas) permettant de définir pour chacun des éléments du processus le nom et la description fonctionnelle.

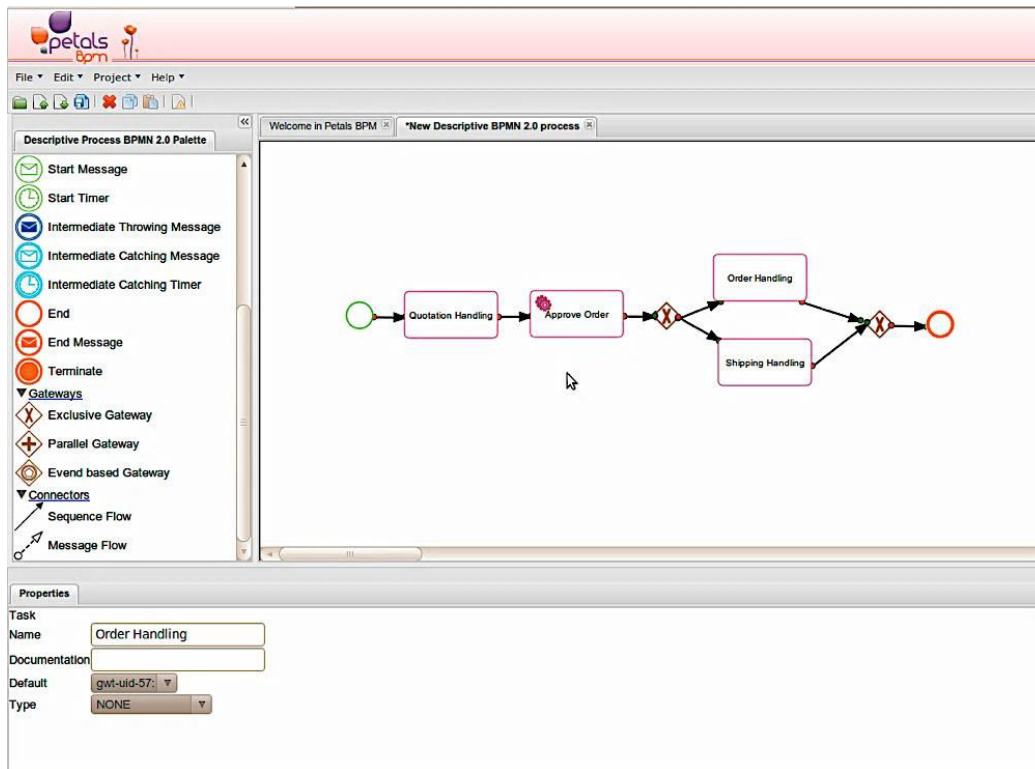


Figure 25 : Interface graphique de Petals BPM.

Comme nous venons de le voir, Petals BPM permet uniquement de décrire fonctionnellement les processus métier collaboratifs. Petals BPM – NFR enrichira la description interne de la palette graphique pour lier automatiquement la modélisation de l'activité BPMN 2.0 au composant « annotation non-fonctionnelle » que nous détaillons ci-après.

III.2.3. Annotation non-fonctionnelle des processus BPMN 2.0

Nous avons souligné dans les chapitres précédents l'importance de la prise en compte des exigences non-fonctionnelles lors de la modélisation des processus métier. Nous avons également mis en évidence l'importance de l'utilisation des standards non seulement pour assurer l'interopérabilité dans ce contexte collaboratif, mais aussi pour l'étendue et l'intégration de nos travaux open-source. Ces deux besoins ont constitué notre point de départ pour la modélisation et l'annotation non-fonctionnelle des processus BPMN 2.0. Dans le chapitre *état de l'art*, nous avons étudié quelques modèles et approches existants (cf. sous-section II.5.4 et II.5.5) et nous avons déduit qu'aucun d'eux ne couvrait à la fois nos deux besoins.

Pour répondre à cette problématique de recherche, nous avons réalisé un travail d'analyse de la littérature sur les exigences non-fonctionnelles qui peuvent potentiellement être définies au niveau métier. Nous effectuons une première analyse de la littérature sur ce sujet en nous basant sur les travaux présentant dans [Santos et al., 2010], [Cysneiros et al., 2004], [Mylopoulos et al., 1992], [Chung et al., 2000], [Rodriguez et al., 2007], [Chung et al., 2009], et [Heinrich et al., 2011]. Chacun de ces travaux fournit un ensemble d'exigences non-fonctionnelles proposé, en extrait, dans le Tableau 3 ci-après (cf. Annexe 2 pour le tableau complet). La dimension horizontale représente la liste des exigences non-fonctionnelles définies dans les travaux de recherche élaborés par les auteurs. Même si les auteurs se différencient sur quelques exigences, nous remarquons que les exigences suivantes : « disponibilité », « prix », « temps de réponse », « sécurité », « exactitude (*Accuracy*) » et « intégrité » sont majoritairement présentes dans tous les travaux. À l'exception de l'exactitude, nous constatons que ces exigences sont également présentes dans le standard WSQF. Nous effectuons une deuxième analyse représentée dans le Tableau 3 (cf. Annexe 2 pour le tableau complet) par la dimension verticale.

Tableau 3 : Extrait du tableau « analyse croisée exigences non-fonctionnelles métier / WSQF ».

		Exigences non-fonctionnelles Métier		WSQF
[Santos et al., 2010]	Performance	Time Performance	Response Time	Response Time
			Throughput	Maximum throughput
	Reliability	Space Performance		
		Availability		Availability
		Fault Tolerance		
	Security	Accuracy		
		Access Control		Authorization + Authentication
		Confidentiality	Internal Confidentiality	Confidentiality
			External Confidentiality	
	Availability		Availability	
...			...	

Légende



Exigence non-fonctionnelle couverte par WSQF



Exigence non-fonctionnelle non couverte par WSQF

Cette analyse ne remet pas en cause notre constatation. Même si la spécification de WSQF est initialement dédiée à la description non-fonctionnelle des services (niveau technique), elle couvre la majorité des exigences non-fonctionnelles proposée dans la littérature pour les processus métier. De plus, comme nous avons employé cette spécification pour l'annotation non-fonctionnelle des services, ceci facilitera la réconciliation le métier et le technique. Ces raisons nous conduisent à la proposition d'utiliser les propriétés du standard WSQF (cf. Annexe 1) pour la modélisation et l'annotation des exigences non-fonctionnelles des activités du processus métier collaboratif BPMN 2.0 [Zribi et al., 2013].

Par ailleurs, comme notre objectif principal est d'associer à chacune des activités un service parmi ceux disponibles dans notre registre EasierGov, il est également nécessaire de savoir la pertinence de chacune des exigences pour le client. Pour cela, *Petals BPM-NFR* fournit aux utilisateurs la possibilité d'exprimer leurs préférences par rapport aux exigences non-fonctionnelles choisies. Cette préférence prend la forme de poids que l'utilisateur attribue à chaque exigence.

Nous schématisons dans la Figure 26, (une reprise de la Figure 24), l'architecture générale de *Petals BPM* à laquelle nous avons rajouté le standard WSQF comme modèle d'annotation non-fonctionnelle et la gestion des préférences des utilisateurs.

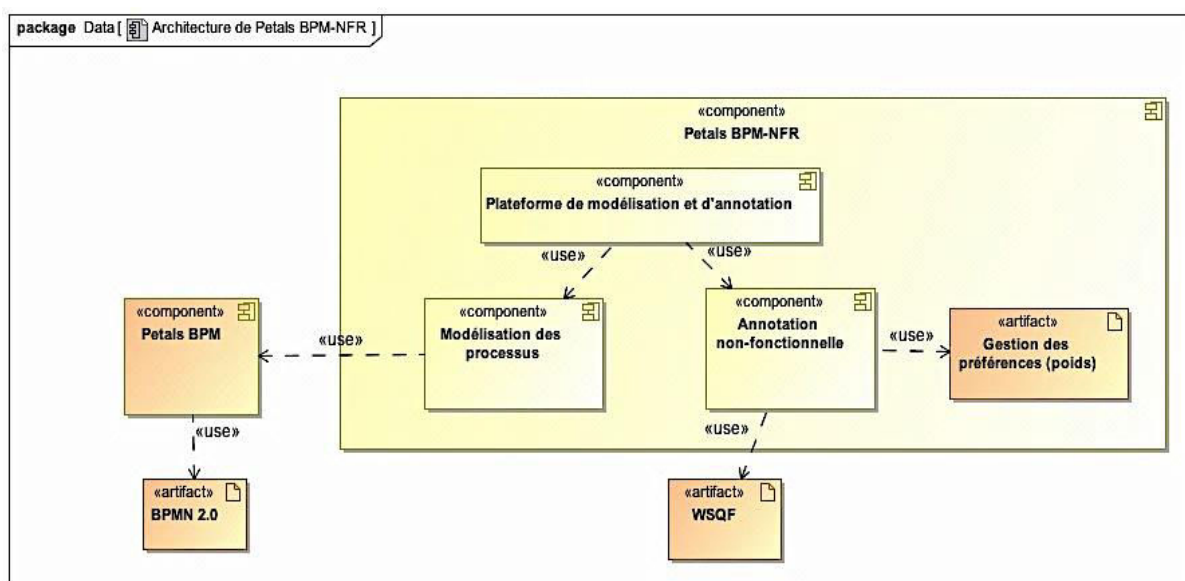


Figure 26 : Architecture de *Petals BPM – NFR* : annotation non-fonctionnelle détaillée.

III.3. EasierGov-NFR : gouvernance SOA pour la gestion des services et des propriétés non-fonctionnelles

La prolifération des services au sein du système d'information rend nécessaire la rationalisation de leur gestion. Par ailleurs, les propriétés non-fonctionnelles sont présentes durant toutes les phases du cycle de vie d'un service. Leur gestion nécessite un registre qui va au delà de cataloguer les données du service et d'en faire l'inventaire. Elle requiert également de stocker les métadonnées des services telles que des contrats de services et les propriétés non-fonctionnelles. La gouvernance SOA permet de répondre à ces problématiques.

EasierGov [Zribi et al., 2012] implémente le paradigme de Gouvernance SOA. Comme l'illustre la Figure 27, il est à la frontière des trois domaines suivants : la Découverte SOA (*SOA Discovery*), la Gestion SOA (*SOA Management*) et les Politiques de la Gouvernance (*Governance Policies*).



Figure 27 : Présentation générale d'EasierGov [Zribi et al., 2012].

- Registre de gouvernance SOA (*SOA Governance Registry*) désigne le registre de services qui assure les fonctionnalités de la découverte du service. Il sert non seulement à faire l'inventaire, à cataloguer les données du service, mais aussi à une base de données pour stocker les métadonnées des services nécessaires à la gouvernance (nous citons à titre d'exemple : *Web Service Description Language (WSDL)*, documents, les propriétés non-fonctionnelles, les contrats de service, etc.). Ce registre permet aux fournisseurs de publier leurs services et les métadonnées respectives. Ces derniers seront mis par la suite à la disposition des utilisateurs pour qu'ils puissent les utiliser.
- Gestion SOA (*SOA Management*) couvre la prise en charge des cycles de vie des services et ce de la phase de conception jusqu'à la phase d'exécution. Ceci inclut la définition des politiques et règles adéquates pour les services à chaque étape de leurs cycles de vie. Par

ailleurs, la gestion SOA comprend aussi la gestion des propriétés non-fonctionnelles des services telles que les contrats de services.

- Politiques de gouvernance (*Governance Policies*) constituent un élément fondamental pour la Gouvernance SOA. En effet, grâce à l'adoption de règles et de standards, la gouvernance assure la compatibilité entre les services de l'organisation et ceux appartenant à d'autres entreprises ou organismes. Les politiques de gouvernance sont définies à chaque étape du cycle de vie et sont fortement liées aux domaines de découverte et de gestion SOA. Chaque acteur impliqué dans la gouvernance a un rôle spécifique et des responsabilités clairement identifiées.

Les travaux de cette thèse portent sur les deux parties : la découverte SOA et la sous-partie gestion des services et leurs propriétés non-fonctionnelles de la partie gestion SOA. Ces parties constitueront les éléments de notre prototype de gouvernance nommé EasierGov-NFR (cf. Figure 28).

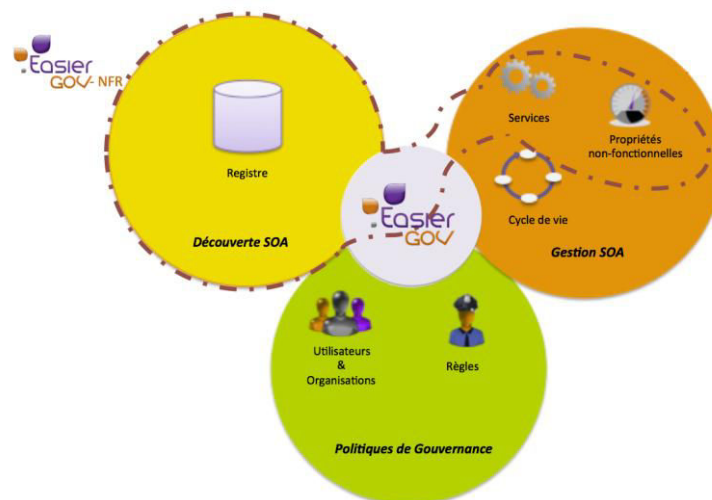


Figure 28 : Présentation générale d'EasierGov-NFR.

III.3.1. Découverte SOA

Au cœur de la gouvernance SOA émerge le concept de registre de services : cet annuaire robuste favorisant la publication, la découverte et la réutilisation des services. Les fonctionnalités de gouvernance, telles que la publication des services, la recherche, la gestion de leurs cycles de vie ou de leurs propriétés non-fonctionnelles sont fondamentales. EasierGov-NFR répond à ces besoins en offrant un registre innovant dédié à la fois aux services et à leurs propriétés non-fonctionnelles. Cette fonctionnalité est essentielle pour faire face à un nombre important de services hétérogènes qui répondent aux mêmes besoins fonctionnels. Par ailleurs, grâce à sa liaison à Petals BPM-NFR, EasierGov-NFR permet d'effectuer la réconciliation non-fonctionnelle entre les activités métier du

processus collaboratif (modélisé et annoté non-fonctionnellement) et les services disponibles. La Figure 29 présente l'architecture générale d'EasierGov-NFR.

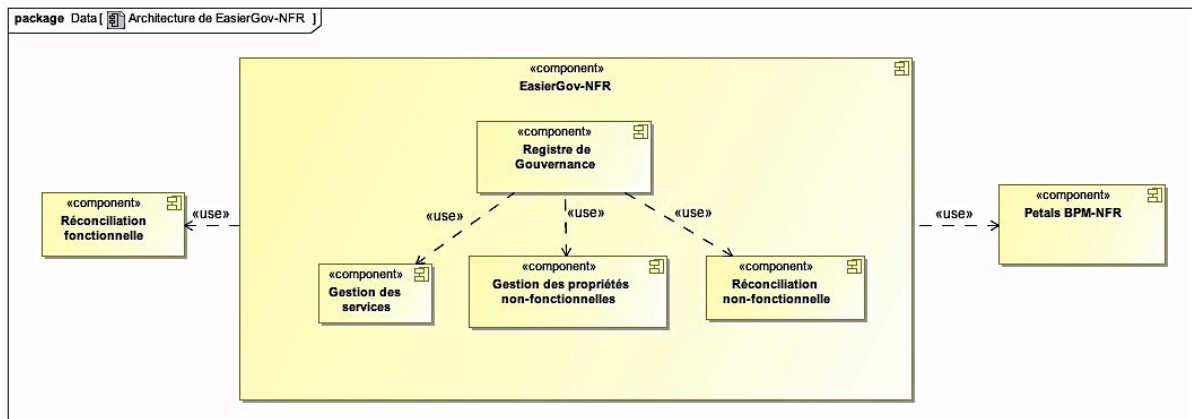


Figure 29 : Présentation de l'architecture d'EasierGov-NFR.

EasierGov-NFR est constitué de trois composants : gestion des services, gestion des propriétés non-fonctionnelles et réconciliation non-fonctionnelle. Les deux premiers sont détaillés dans les sections qui suivent et le troisième dans le chapitre suivant (cf. Chapitre IV). Comme nous l'avons détaillé depuis le début de nos travaux, la réconciliation non-fonctionnelle tente de sélectionner les services (parmi ceux fournis par la réconciliation fonctionnelle) pour les activités métier du processus modélisé dans Petals BPM-NFR. Ceci explique l'utilisation de ces deux composants (réconciliation fonctionnelle et Petals BPM-NFR) par EasierGov-NFR.

III.3.2. Gestion des services

Pour assurer le bon fonctionnement d'un système d'information et une meilleure rationalisation de la gestion des services, il est nécessaire d'appliquer des politiques adéquates. Ces politiques sont définies à chaque étape du cycle de vie par le responsable de gouvernance d'EasierGov. La gestion des services dans EasierGov-NFR commence à partir de la phase de publication. Afin d'aborder la problématique d'hétérogénéité de services métier, une des politiques qu'il est important de définir est la définition d'un langage commun pour la gestion des descriptions des services dans le registre de gouvernance. À cette fin, nous adoptons le langage standard et communément utilisé WSDL (cf. Chapitre II, sous-section II.2.2).

Dans EasierGov-NFR et pour la gestion des services, nous identifions deux rôles : celui de fournisseur de services et celui de consommateur. Ces deux rôles sont menés à suivre les politiques définies dans le registre de gouvernance. Les diagrammes de cas d'utilisation ci-après (cf. Figures 30 et 31) décrivent les fonctionnalités de chacun d'eux.

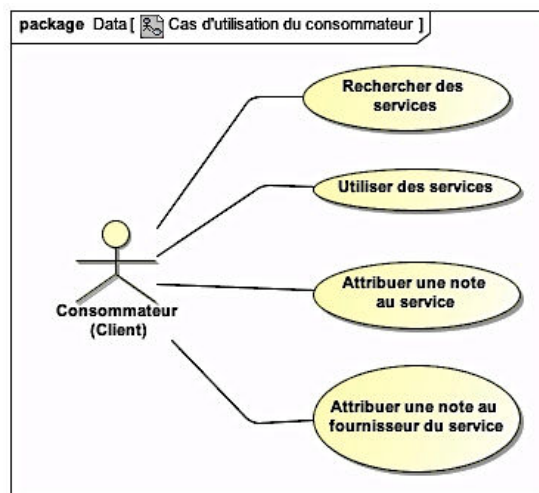
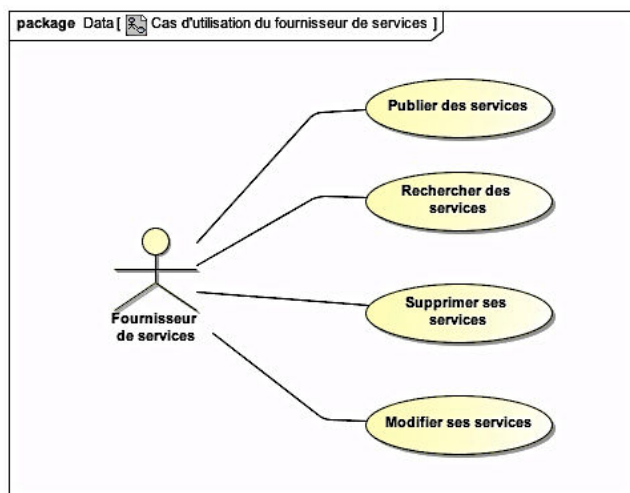


Figure 30 : Cas d'utilisation du fournisseur de service. **Figure 31 :** Cas d'utilisation du consommateur.

Le *fournisseur de services* après avoir conçu ses services, implémenté leurs fonctionnalités en suivant le standard WSDL et effectué des tests, peut les publier dans le registre de gouvernance EasierGov-NFR. Il peut également rechercher des services. En outre, le registre lui permet aussi de gérer ses services après leur publication en lui attribuant le droit de modification et de suppression.

Le *consommateur* (client) peut rechercher des services dans le registre de gouvernance et notamment les utiliser. Par la suite, le consommateur a la possibilité d'exprimer sa satisfaction par rapport à l'utilisation du service et peut le noter et/ou noter son fournisseur.

III.3.3. Gestion des propriétés non-fonctionnelles

Les services Web peuvent fonctionner en se basant uniquement sur leurs besoins fonctionnels. Cependant, pour plus d'efficacité, un service doit également fournir une certaine garantie quant à sa disponibilité, sa fiabilité, son efficacité et donc quant à ses propriétés non-fonctionnelles. Ces propriétés sont très importantes pour garantir et maintenir la confiance auprès des clients. Dans cette sous-section, nous présentons la manière avec laquelle nous pouvons gérer les propriétés non-fonctionnelles à partir du registre de gouvernance SOA : EasierGov-NFR.

III.3.3.1. Annotation non-fonctionnelle des services

La spécification de WSQF permet d'enrichir la description de services par des propriétés non-fonctionnelles. Toutefois pour pouvoir les gérer, il est bien nécessaire de savoir où nous pouvons les trouver.

Le contrat de service (appelé aussi l'accord de niveau de service, *Service Level Agreement – SLA*) représente un accord bilatéral entre le fournisseur de service et le client. Cet accord spécifie et

identifie explicitement (i) les attentes, (ii) les engagements des différentes parties autour de l'utilisation du service en terme de qualité de service et (iii) les pénalités en cas de manquement ou les récompenses en cas d'une meilleure prestation. En effet, sans contrat défini explicitement ou implicitement, il est difficile de parler de la qualité de service et donc des propriétés non-fonctionnelles [Mammeri, 2005].

Un contrat SLA est composé généralement de trois sections [Keller et al., 2003] : (i) la première section, intitulée « *Parties* », spécifie les parties impliquées dans le contrat, à savoir : les signataires (le fournisseur de service et le client) et les parties tierces qui participent au contrôle du contrat, (ii) la deuxième section est *Service « Description »* qui définit la description de service. Cette dernière contient les opérations du service, leurs messages, l'encodage du transport de ces messages, les propriétés non-fonctionnelles, la durée et la fréquence des mesures des qualités de service. Enfin, (iii) la troisième section « *Obligation* » comprend les obligations du contrat, à savoir : la période de validité, les contraintes à respecter, les pénalités en cas de sa violation et les récompenses.

Tout comme les services, les contrats SLA ont un cycle de vie. Celui appliqué à cette architecture est composé de six phases [Wustenhoff et al., 2002] et [Sun et al., 2005] :

1. *Établissement d'un modèle SLA* (appelé aussi un gabarit ou un *template*) : cette première phase consiste en la spécification, la modélisation et la publication d'un modèle de contrat SLA ;
2. *Négociation du contrat* : une fois que le service est choisi, il est important de spécifier le niveau de qualité requis ou de négocier les clauses existantes dans le modèle fourni par le fournisseur ;
3. *Etablissement du contrat* : dans cette phase, le contrat SLA est créé et doit maintenant être signé par le fournisseur du service et le client ;
4. *Exécution et surveillance des performances engagées* : cette phase consiste à surveiller l'exécution du service afin de s'assurer que toutes les clauses engagées ont été bien respectées par les deux parties ;
5. *Terminaison* : cette phase correspond à la fin de l'exécution du service ;
6. *Application des pénalités ou récompenses* : en cas de non respect et de violation des clauses du contrat, il est important d'appliquer les sanctions prédéfinies ou dans le cas contraire les récompenses engagées.

Il est évident que l'ensemble du cycle de vie d'un contrat de service est basé sur un modèle utilisé pour représenter les différentes parties de ses clauses. Ce modèle contient la liste des services offerts et intègre les capacités du fournisseur en terme de qualité de service, et donc en terme des propriétés non-fonctionnelles, auxquelles son service prétend d'assurer [Sun et al., 2005].

Il existe plusieurs spécifications pour la représentation de ce modèle SLA. Parmi lesquelles, nous citons WSOL (*Web Service Offerings Language*) [Tosic et al., 2002], WSLA (*Web Service Level Agreement*) [Keller et al., 2003] et WS-Agreement (*Web Service-Agreement*) [Andrieux et al., 2007]. Elles s'intéressent toutes à la définition du contrat SLA. Toutefois, le standard WS-Agreement se focalise plus sur la création des modèles de contrat. Il se base sur le langage XML et comporte une spécification claire, flexible et assez simple. Par ailleurs, ce standard est extensible à tout modèle écrit en XML. De plus, l'équipe R&D de Linagora Labs a réalisé un outil de surveillance (*monitoring*) appelé « EasierBSM » permettant le contrôle de l'exécution des contrats de type WS-Agreement pour vérifier leur bon déroulement et leur non-violation. Ceci nous a conduits à l'adopter et à faire l'hypothèse que les fournisseurs créent leurs SLA templates dans le respect de la spécification WS-Agreement.

III.3.3.2. Modèle de contrat de service Web (*WS-Agreement template*)

Un template SLA, selon la spécification de WS-Agreement, est effectué par le fournisseur de services et est composé de trois grandes parties. Nous résumons sa structure dans la Figure 32 ci-après.

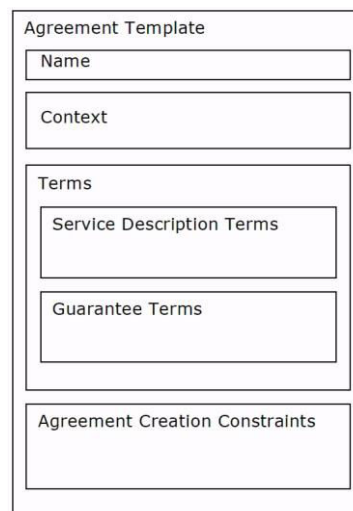


Figure 32 : Structure d'un template SLA selon WS-Agreement [Andrieux et al., 2007].

- *Name et Context* : la première section est optionnelle et définit le nom du template SLA. Le contexte contient l'ensemble des métadonnées du contrat telles que : le fournisseur, la date d'expiration du template, etc. ;
- *Termes* : cette section contient les termes qui décrivent le contrat. Ils englobent deux sous-parties : *Service Description Terms* (SDT) et *Guarantee Terms*. Les modalités de description de service (SDT) spécifient les descriptions fonctionnelles du service à fournir. Les modalités de garantie (*Guarantee Terms*) incluent les capacités en termes de qualité de

services (propriétés non-fonctionnelles) auxquelles le service peut répondre, les pénalités ou / et les récompenses ;

- *Agreement Creation Constraints* : les contraintes de création de l'accord définissent les règles supplémentaires qui doivent être suivies lors de la proposition d'un contrat par le client.

Les propriétés non-fonctionnelles sont donc définies dans les modalités de garantie. Ces dernières sont composées de quatre éléments : *Service Scope*, *Qualifying Condition*, *Service Level Objectives* et *Business Value List*.

La portée du service (*Service Scope*) spécifie le service qui est couvert par les modalités de garantie. La condition de qualification (*Qualifying Condition*) précise les conditions préalables qui doivent être remplies pour l'application d'une garantie. Les objectifs de niveau de services (*Service Level Objectives - SLO*) définissent les propriétés non-fonctionnelles qui sont assurées par le service et qui doivent être monitorées afin de déterminer la réalisation des modalités de garanties. Enfin, la liste des valeurs métier (*Business Value List*) comprend les pénalités, les récompenses et les propriétés non-fonctionnelles qui ne nécessitent pas d'être surveillées (e.g. le prix).

Comme nous pouvons le remarquer, ce sont les sous-parties *SLO* et *Business Value List* de la partie *Guarantee Terms* qui permettent au fournisseur du service de définir les propriétés non-fonctionnelles auxquelles son service prétend de répondre. Afin de déterminer quelles propriétés peuvent être définies dans chacune de sous-parties, nous proposons dans la sous-section qui suit une classification des critères non-fonctionnels de WSQF.

III.3.3.3. Classification des propriétés non-fonctionnelles

Après l'étude et l'analyse de l'ensemble des propriétés non-fonctionnelles définies dans la spécification de WSQF, nous distinguons trois catégories distinctes : des *variables Fournisseur*, des *variables Client* et des *variables analysées*.

- Les *variables Fournisseur* sont définies dans le template SLA par le fournisseur du service. Nous les avons appelées ainsi étant donné que seul le fournisseur a le droit de gérer (ajouter, supprimer et modifier) les valeurs de ces propriétés. Nous décomposons ces variables en deux sous-catégories : des *variables monitorables* et d'autres qui sont *non monitorables*. La première sous-catégorie regroupe les propriétés non-fonctionnelles qui nécessitent d'être monitorées lors de l'exécution du service telles que le temps de réponse, la disponibilité, le cryptage, les propriétés d'interopérabilité, etc. Ceci nous permettra de vérifier que les valeurs des propriétés non-fonctionnelles engagées par le fournisseur ont bien été respectées ou non. Contrairement à ces variables, les valeurs des *variables non*

monitorables ne nécessitent pas d'être vérifiées à l'exécution telles que le prix, les pénalités, les récompenses, etc.

- Les valeurs des *variables Client* sont renseignées directement par le client du service. Par exemple, à l'issue de l'utilisation d'un service, un client peut attribuer une note à lui ou à son fournisseur. Le degré de satisfaction contribue à constituer ce qui est communément appelé, la réputation d'un service ou d'un fournisseur ;
- Les valeurs des *variables analysées* ne sont introduites ni par le fournisseur du service ni par le client. Elles sont renseignées par le registre de gouvernance. Nous retrouvons dans cette catégorie la propriété popularité du service (*Service Recognition*). Nous considérons que la valeur de cette propriété est calculée selon le nombre de clients qui ont utilisé le service.

Nous avons classé, dans le Tableau 4 ci-après, l'ensemble des propriétés non-fonctionnelles de WSQF (la dimension horizontale) dans chacune de ces catégories/sous-catégories (la dimension verticale).

Tableau 4 : Classification générale des propriétés non-fonctionnelles de WSQF.

		Variables Fournisseur		Variables Client	Variables analysées	
		Variables monitorables	Variables non monitorables			
Propriétés non-fonctionnelles de WSQF	Business	Price		X		
		Penalty and Incentive		X		
	Value	Business Performance		X		
		Service Reputation			X	
	Quality	Service Provider Reputation			X	
		Service Recognition				X
	Service	Response Time	X			
		Maximum Throughput	X			
	Level	Availability	X			
		Successability	X			
	Measurement	Accessibility	X			
		Standard Adoptability	X			
	Interoperability	Standard Conformability	X			
		Relative Proofness	X			
	Business	Message Reliability	X			
		Transaction Integrity	X			
	Processing	Collaborability	X			
		Informability	X			
	Manageability	Observability	X			
		Controllability	X			
	Security	Encryption	X			
		Non Repudiation	X			
		Authentication	X			
		Authorization	X			
		Availability	X			
		Audit	X			
		Integrity	X			
	Privacy	X				

Étant donné que le modèle de contrat du service est conçu par le fournisseur, seules les propriétés non-fonctionnelles de WSQF de la catégorie *variables Fournisseur* peuvent y être définies. Comme nous l'avons détaillé dans la sous-section précédente (cf. III.3.3.2), à l'inverse des assertions *business value list*, les assertions *SLO* contiennent les propriétés qui nécessitent d'être surveillées et vérifiées lors de l'exécution afin de s'assurer de leur conformité. Nous proposons dans le Tableau 5 un résumé des propriétés non-fonctionnelles du type *variables Fournisseur* qui peuvent être incluses dans chacune des deux parties (SLO ou business value list) du modèle de contrat WS-Agreement.

Tableau 5 : Extension de WS-Agreement par les propriétés non-fonctionnelles de WSQF de type variables Fournisseur.

			Modèle de contrat WS-Agreement	
			SLO	Business value list
Propriétés non-fonctionnelles de WSQF de type Variables Fournisseur	Business Value	Price		X
		Penalty and Incentive		X
		Business Performance		X
	Service Level Measurement	Response Time	X	
		Maximum Throughput	X	
		Availability	X	
		Successability	X	
		Accessibility	X	
	Interoperability	Standard Adoptability	X	
		Standard Conformability	X	
		Relative Proofness	X	
	Business Processing	Message Reliability	X	
		Transaction Integrity	X	
		Collaborability	X	
	Manageability	Informability	X	
		Observability	X	
		Controllability	X	
	Security	Encryption	X	
		Non Repudiation	X	
		Authentication	X	
		Authorization	X	
Availability		X		
Audit		X		
Integrity		X		
Privacy	X			

III.3.3.4. Gestion des propriétés non-fonctionnelles dans EasierGov-NFR

Dans le registre de gouvernance EasierGov-NFR, outre les fonctionnalités de découverte et de publication de services nous proposons des fonctionnalités dédiées à la gestion des propriétés non-fonctionnelles. Cette gestion couvre la prise en compte de l'ensemble des propriétés du standard WSQF pour chaque service publié par son fournisseur.

➤ *La gestion des variables Fournisseur*

Nous proposons dans EasierGov-NFR une librairie permettant aux fournisseurs de publier, pour un service donné, un ou plusieurs modèles de contrat de type WS-Agreement. Ces modèles, que nous avons étendus par WSQF, contiennent les propriétés non-fonctionnelles de type variables Fournisseur. Il est donc nécessaire de pouvoir récupérer leurs valeurs afin de les enregistrer dans notre registre de gouvernance SOA. Pour répondre à ceci, nous proposons les trois étapes suivantes :

1. Publication des services : EasierGov-NFR contient une librairie permettant aux fournisseurs de publier les descriptions de leurs services de type WSDL ;
2. Publication des templates SLA : à l'issue de la phase de publication du service, le registre de gouvernance permet au fournisseur de publier également un ou plusieurs modèles de contrats SLA de type WS-Agreement associés à ce service ;
3. Extraction dynamique et enregistrement des valeurs des propriétés non-fonctionnelles : pour chaque template SLA publié, EasierGov-NFR extrait dynamiquement les valeurs des propriétés non-fonctionnelles et les stocke dans le registre.

➤ *La gestion des variables Client*

Afin de gérer les *variables Client*, à savoir la réputation du service et la réputation du fournisseur de services, nous considérons le déroulement suivant (cf. Figure 33) :

Tout commence par l'étape de découverte du service. EasierGov-NFR fournit aux clients un accès vers la librairie de gestion de services (*services-api*) afin de leur permettre de rechercher un ou plusieurs services parmi ceux existants. Après avoir sélectionné le service qui est susceptible de répondre à ses attentes, deux possibilités se présentent :

- le client utilise le service et par la suite il peut lui attribuer une note ou à son fournisseur selon le degré de sa satisfaction ;
- le client note directement le service ou son fournisseur (par exemple dans le cas où il a préalablement utilisé le service, mais il a oublié de laisser une évaluation).

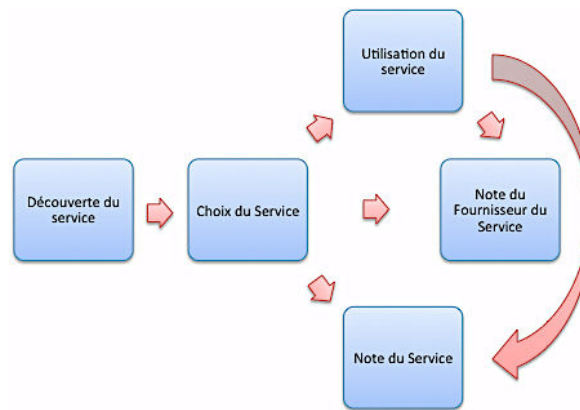


Figure 33 : Gestion des variables non mesurables : Variables Client.

➤ **La gestion des variables analysées**

Dans notre modèle WSQF, seule la variable *popularité du service* appartient à cette catégorie. Afin de la prendre en compte, nous considérons dans EasierGov-NFR qu'elle est calculée selon le nombre de clients qui ont utilisé le service en question.

III.4. Conclusion

Dans ce chapitre, nous avons présenté notre contribution dans le domaine de l'annotation non-fonctionnelle des processus métier BPMN 2.0 et l'apport de la gouvernance SOA dans la gestion des services et de leurs propriétés non-fonctionnelles. Nous avons mis en évidence les problématiques à résoudre et nous avons présenté :

- notre plateforme Petals BPM-NFR pour l'enrichissement de la modélisation graphique des processus métier BPMN 2.0 par l'ajout de l'aspect non-fonctionnel. Nous avons commencé par une analyse de la littérature des exigences non-fonctionnelles qui peuvent être définies au niveau métier. Par la suite, nous avons montré, la possibilité d'utiliser le standard WSQF (destiné à l'origine pour une utilisation technique) au niveau métier. Enfin, nous avons présenté l'architecture et les fonctionnalités de la plateforme Petals BPM-NFR.
- notre cadre de gouvernance SOA EasierGov-NFR qui définit un registre de services qui va au delà de cataloguer les services. Il permet également de gérer les propriétés non-fonctionnelles des services et les contrats de services. D'abord, nous avons présenté l'architecture globale du registre. Ensuite, nous avons détaillé la gestion des services dans EasierGov-NFR. Enfin, nous nous sommes intéressés à la partie relevant de la gestion des propriétés non-fonctionnelles et des contrats de services.

Dans le chapitre suivant, nous détaillons la réconciliation des activités métier du processus collaboratif BPMN 2.0 annotées à l'aide d'exigences non-fonctionnelles en assignant à chacune de ces activités un service (parmi ceux publiés dans le registre EasierGov-NFR et qui satisfont les attentes fonctionnelles) qui répond à ses exigences non-fonctionnelles.